White Paper Report

Report ID: 101015

Application Number: HD5115310

Project Director: Haimonti Dutta (hd2200@columbia.edu)

Institution: Columbia University

Reporting Period: 9/1/2010-2/28/2013

Report Due: 5/31/2013

Date Submitted: 5/31/2013

Leveraging "The Wisdom of the Crowds" for Efficient Tagging and Retrieval of Documents from the Historic Newspaper Archive of the New York Public Library

Haimonti Dutta, William Chan, Manoj Pooleery and Megha Gupta

Final Report submitted to the
National Endowment of Humanities

Table of Contents

1 Introduction **1**

2 Background **4**

2.1 The Data 4

3 The Bodhi System **8**

3.0.1 System Architecture 8

3.0.2 The Database Design 10

3.0.3 The UI Design 11

4 Topic Modeling **13**

4.1 Ground Truth 13

4.2 Community Structure Detection 14

4.2.1 Community Structure Detection Using Modularity Maximization: . 15

4.3 Second Opinion 16

4.3.1 Case Study 1: Unpaid Annotators 16

4.3.2 Case Study 2: Mechanical Turkers as Annotators 19

4.4 Information Retrieval Metrics 22

4.5 Experimental Results 24

4.6 Graphical Models for Community Structure Discovery in Document Networks 27

4.6.1 Terminology 27

5 Grant Products **30**

I Appendices **32**

5.1 Appendix 33

II Bibliography **34**

Bibliography **35**

Chapter 1

Introduction

Chronicling America[1] is a website that provides access to over 3 million historic newspapers scanned by the National Digital Newspaper Program (NDNP). It is an initiative of the National Endowment for Humanities (NEH) and the Library of Congress (LC) and its goal is to develop an online, searchable database of historically significant newspapers between 1836 and 1922. State libraries, historical societies and universities have been funded by the NEH to generate scanned newspaper pages representing the state's regional history, geographic coverage, and events which will be archived by the LC. The New York Public Library (NYPL) is part of this initiative and has scanned 200,000 newspaper pages published between 1860 and 1920 from microfilm.

While the images scanned by the NYPL are sent to the Library of Congress for hosting on the Chronicling America website, duplicate copies of the archive are also available at NYPL for use by its patrons. These digitized pages offer a wealth of data for researchers, historians, genealogists and other patrons of the library. For example, the opening of the Brooklyn Bridge in 1883, the construction of an immigration station at Ellis Island in 1890, the historic opening of the Metropolitan Opera House on Broadway and 39th Street are some interesting local news items of the time. Other topics widely covered by the American press include presidential administrations (Cleveland (1885 – 1894), Garfield (1880 - 1883), McKinley (1897 - 1901), Theodore Roosevelt (1901 - 1912)), natural calamities (Galveston

[1] `http://chroniclingamerica.loc.gov/`

flood of 1900, San Francisco earthquake 1906), the sinking of the Titanic, events pertaining to the first world war and news from the world of medicine (patented medicines, spread of epidemics, new discoveries). To effectively use this archive, developing sophisticated search and retrieval mechanisms is crucial.
In order to make a newspaper available for searching on the Internet, the following processes must take place:

1. The microfilm copy or original paper is scanned.
2. Master and Web image files are generated.
3. Metadata is assigned for each page to improve the search capability of the newspaper.
4. Optical Character Recognition (OCR) software is run over high resolution images to create searchable full text, and
5. OCR text, images, and metadata are imported into a digital library software program.

The newspaper archives can currently be searched using the OpenSearch protocol[2]. Unfortunately, these search facilities are rudimentary and irrelevant documents are often more highly ranked than relevant ones. For instance consider a search for a natural calamity like the April 18th, 1906 San Francisco earthquake which killed approximately 2000 people and measured 7.8 on the Richter scale; if the keywords "earthquake San Francisco" are entered as the search criteria in the digitized newspaper archive along with a date range 01/01/1906 to 12/31/1906, the first document returned is Page 7, April 19th of the 1906 Los Angeles Herald with an article "Fifty people killed at San Jose" (the word "San" is tagged); the second document returned is the June 3rd, 1906 issue of "The San Francisco Sunday Call" with a full-page illustration of a drama by Frederick Irons Bamford and the third document is page 13, June 3rd, 1906 issue of "The Sunday Call" with a cartoon of "Major ozone′s fresh air crusade". The retrieval technique missed finding Page 1, April 19th, 1906 of the newspaper "The Sun" published from New York, which had a headline article "Earthquake lays Frisco in ruins".

[2] `http://www.opensearch.org/Home`

On investigating the reasons why irrelevant documents are ranked higher, it was found that the newspapers are scanned on a page-by-page basis and article level segmentation is poor or non-existent. The OCR scanning process is far from perfect and the documents generated from it contains a large amount of garbled text. In addition, categorization of article level data using the OCR software was not very successful – most of the articles are labeled "editorial" and there is no fine grained classification into categories such as crime, politics, medicine, etc. For example, an attempt to categorize articles in the edition of *The Sun* newspaper published on November 4th, 1894 resulted in 338 articles classified as editorial, 32 unclassified[3], 10 sports, 23 advertising, 5 commercial, 3 birth-related announcements, and 2 reviews.

Even though the New York Public Library has put in substantial effort into improving the quality of the images and text obtained from OCR by testing each word scanned against english dictionaries, manually re-typing newspaper headlines and applying categories to articles – the digital outputs from the OCR software are not good enough to ensure adequate quality of text retrieval or to meet user expectations. Consequently, we conjectured that a crowdsourcing project involving patrons of the library who use the archive frequently for research and learning could assist the task of categorizing articles of this huge online repository.

This white paper is organized as follows: Chapter 2.1 describes the characteristics of the data, Chapter 3 provides details of the prototype system BODHI built for OCR text correction and tagging, Chapter 4 describes algorithms for categorizing newspaper articles based on topics, Chapter 5 describes the products developed on this grant.

[3]These were later identified as banners of the newspaper.

Chapter 2

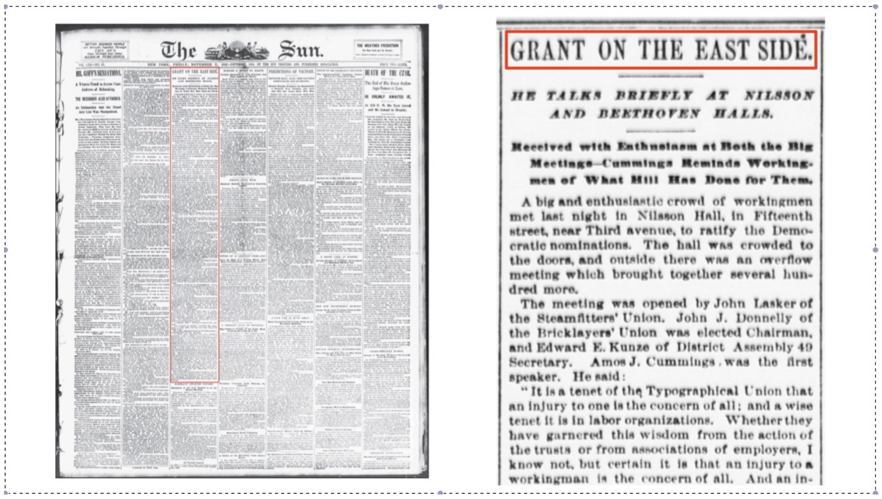
Background

2.1 The Data

Figure 2.1 shows a scanned newspaper page (*The Sun*, November 2, 1894) from the NYPL archive and an article from this paper. The historical newspaper archive contains two types of XML files: (1) **Page-Level XMLs:** For each page of a newspaper, there is an XML file that contains metadata about the page and the text in it. Each word scanned is stored as a string in the page level XML (See Figure 2.2) along with possible alternative suggestions for the word[1]. The text is extracted from the page level XML using Xpath queries and stored in a PostGreSQL database. (2) **Issue-Level XMLs:** The issue-level XMLs (illustrated in Table 2.3) provide the following information about articles: (a) *Headlines cleaned by humans* which are of much higher quality than the text produced by the OCR software. (b) *Article segmentation information:* Each newspaper article is represented as a collection of one or more text blocks whose pixel coordinates are available. This helps to determine where one article ends and the next one begins and is particularly useful when an article spans more than one page. (c) *High-level categorization* of the articles produced by the OCR software. We have access to only a subset of the NYPL archive[2] – issues of *The Sun* newspaper from November 1, 1894 to December 31, 1894. (d) In addition, issue level XMLs also store the date of publication, volume number and issue number and provide pointers

[1]We found that its primary selections are usually better than their alternatives.

[2]These have been substantially cleaned by humans



GRANT ON THE EAST SIDE.

HE TALKS BRIEFLY AT NILSSON AND BEETHOVEN HALLS.

Received with Enthusiasm at Both the Big Meetings—Cummings Reminds Workingmen of What Hill Has Done for Them.

A big and enthusiastic crowd of workingmen met last night in Nilsson Hall, in Fifteenth street, near Third avenue, to ratify the Democratic nominations. The hall was crowded to the doors, and outside there was an overflow meeting which brought together several hundred more.

The meeting was opened by John Lasker of the Steamfitters' Union. John J. Donnelly of the Bricklayers' Union was elected Chairman, and Edward E. Kunze of District Assembly 49 Secretary. Amos J. Cummings was the first speaker. He said:

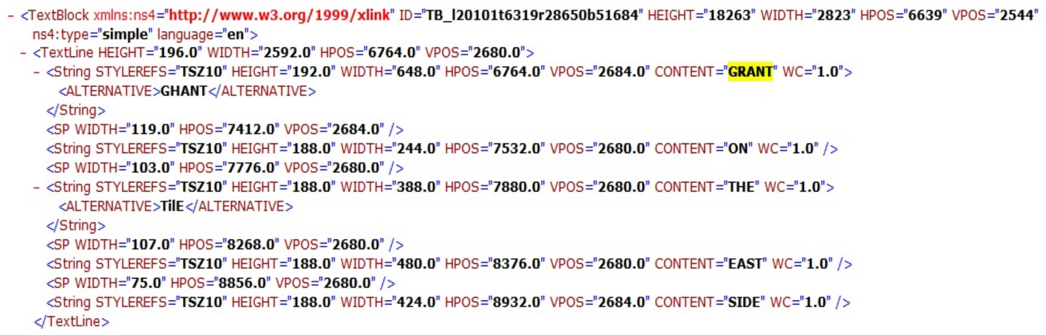
"It is a tenet of the Typographical Union that an injury to one is the concern of all; and a wise tenet it is in labor organizations. Whether they have garnered this wisdom from the action of the trusts or from associations of employers, I know not, but certain it is that an injury to a workingman is the concern of all. And an in-

Figure 2.1: (Left) A newspaper page from the NYPL archive. The red-border shows an article from the newspaper, zoomed in on the right hand figure.

to the location and names of the page-level XML files.

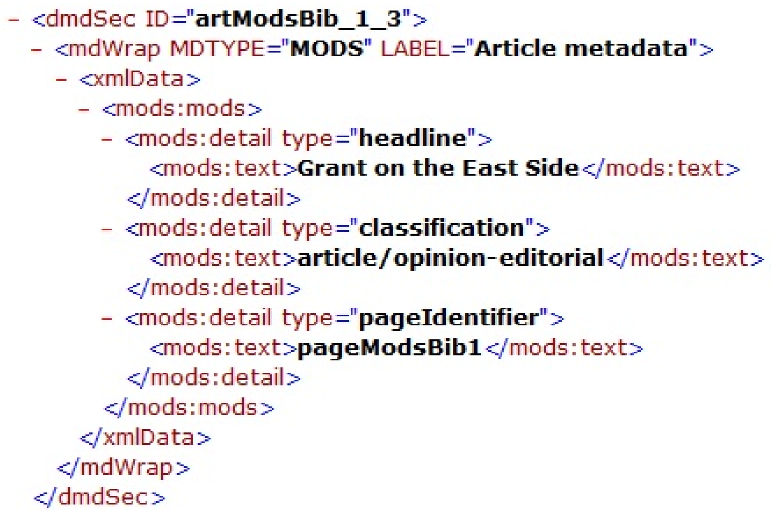
Table 2.4 shows all the categories found by the OCR software for "The Sun" newspaper published between November 2nd, 1894 – December 31st, 1894. Articles in the "editorial/opinion" and "sports" categories contain statistically significant amounts of text while reviews, illustrations, birth/death/wedding announcements are not included in our study.

Pre-processing: We preprocess the documents to reduce dimensionality and have clean data to learn from. For each article, a bag-of-words representation and tf-idf weights are obtained. Stop words such as "the", "and", etc. are removed from the set of words. Letters of length three or less and words that contain digits or repeated characters (e.g. "paaa" and "ornnn") are also removed. After applying the above noise reduction techniques, the dimensionality of the feature space is 3210.



```
- <TextBlock xmlns:ns4="http://www.w3.org/1999/xlink" ID="TB_l20101t6319r28650b51684" HEIGHT="18263" WIDTH="2823" HPOS="6639" VPOS="2544"
   ns4:type="simple" language="en">
 - <TextLine HEIGHT="196.0" WIDTH="2592.0" HPOS="6764.0" VPOS="2680.0">
  - <String STYLEREFS="TSZ10" HEIGHT="192.0" WIDTH="648.0" HPOS="6764.0" VPOS="2684.0" CONTENT="GRANT" WC="1.0">
      <ALTERNATIVE>GHANT</ALTERNATIVE>
    </String>
    <SP WIDTH="119.0" HPOS="7412.0" VPOS="2684.0" />
    <String STYLEREFS="TSZ10" HEIGHT="188.0" WIDTH="244.0" HPOS="7532.0" VPOS="2680.0" CONTENT="ON" WC="1.0" />
    <SP WIDTH="103.0" HPOS="7776.0" VPOS="2680.0" />
  - <String STYLEREFS="TSZ10" HEIGHT="188.0" WIDTH="388.0" HPOS="7880.0" VPOS="2680.0" CONTENT="THE" WC="1.0">
      <ALTERNATIVE>TilE</ALTERNATIVE>
    </String>
    <SP WIDTH="107.0" HPOS="8268.0" VPOS="2680.0" />
    <String STYLEREFS="TSZ10" HEIGHT="188.0" WIDTH="480.0" HPOS="8376.0" VPOS="2680.0" CONTENT="EAST" WC="1.0" />
    <SP WIDTH="75.0" HPOS="8856.0" VPOS="2680.0" />
    <String STYLEREFS="TSZ10" HEIGHT="188.0" WIDTH="424.0" HPOS="8932.0" VPOS="2684.0" CONTENT="SIDE" WC="1.0" />
  </TextLine>
```

Figure 2.2: Page Level XML file showing the article with headline "GRANT ON THE EAST SIDE".



```
- <dmdSec ID="artModsBib_1_3">
 - <mdWrap MDTYPE="MODS" LABEL="Article metadata">
  - <xmlData>
   - <mods:mods>
    - <mods:detail type="headline">
        <mods:text>Grant on the East Side</mods:text>
      </mods:detail>
    - <mods:detail type="classification">
        <mods:text>article/opinion-editorial</mods:text>
      </mods:detail>
    - <mods:detail type="pageIdentifier">
        <mods:text>pageModsBib1</mods:text>
      </mods:detail>
     </mods:mods>
    </xmlData>
   </mdWrap>
  </dmdSec>
```

Figure 2.3: A segment of the issue-level XML file illustrating the OCR Classification (as "article/editorial") for the article and its headline.

Category	Article Counts
Editorial/Opinion	11,441
Sports	764
Advertising	683
Commercial/Legal/Public notices	361
Birth/Death/Wedding	158
Reviews	45
Illustrations	2
Unclassified	785
Total	14,239

Figure 2.4: Top-level categories of articles from OCR software for *The Sun* newspaper between November 2nd, 1894 and December 31st, 1894.

Chapter 3

The Bodhi System

To enable tagging and OCR correction, we designed a prototype system called "BODHI"[1] that allows patrons to correct garbled OCR text, enter keywords for tagging articles, provide useful information about segmentation of the article (for example – is the article continued onto another page or not), links to other related articles, etc. These user-provided meta-data will augment the scanned text and image data obtained from the OCR scanning process. While the Chronicling America website has many modules to manage the newspaper digitization workflow, the BODHI system focuses only on improving search and retrieval.

3.0.1 System Architecture

The prototype was developed and tested on a small sample of data (newspaper articles from November and December of "The Sun", published from New York in 1894). The headlines of these newspaper articles have been cleaned by students from a high school in New York as part of the data cleaning project undertaken by the New York Public Library. This data was made available to the CCLS researchers for development of the prototype system and research on topic modeling.

[1]The word Bodhi in Sanskrit and Pali means the understanding possessed by a Buddha regarding the nature of things. Translated to English it means enlightment.

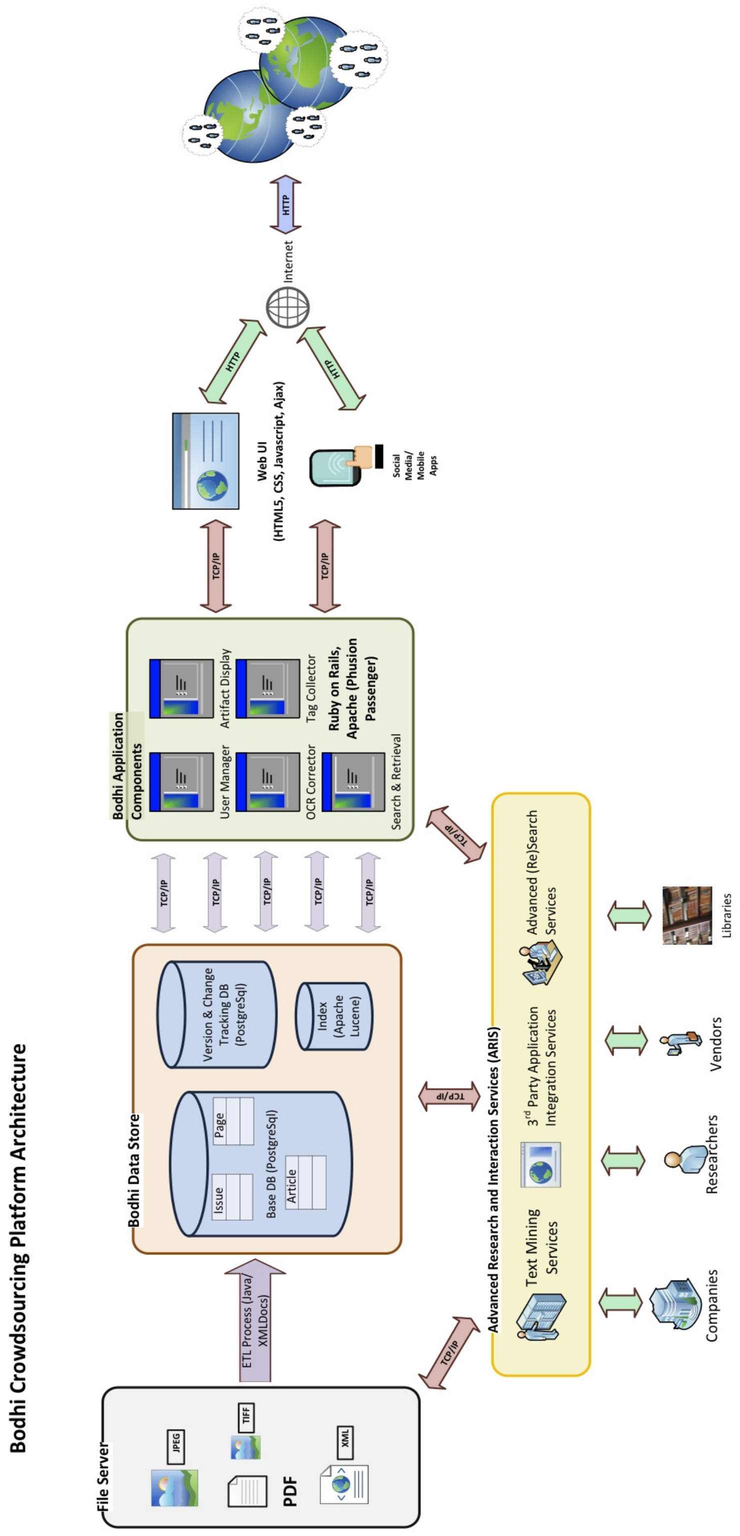

Bodhi Crowdsourcing Platform Architecture
File Server
JPEG
TIFF
PDF
XML
ETL Process (Java/ XMLDocs)
Bodhi Data Store
Issue
Page
Article
Base DB (PostgreSql)
Version & Change Tracking DB (PostgreSql)
Index (Apache Lucene)
TCP/IP
Bodhi Application Components
User Manager
Artifact Display
OCR Corrector
Tag Collector
Search & Retrieval
Ruby on Rails, Apache (Phusion Passenger)
Web UI (HTML5, CSS, Javascript, Ajax)
Social Media/ Mobile Apps
HTTP
Internet
Advanced Research and Interaction Services (ARIS)
Text Mining Services
3rd Party Application Integration Services
Advanced (Re)Search Services
Companies
Researchers
Vendors
Libraries

Figure 3.1: Architecture and Implementation Details of the BODHI Crowd Sourcing System

3.0.2 The Database Design

Figure 3.1 presents the architecture and digital technology that was used in development of the "BODHI" system. The file server stores the digitized pages of the newspaper as jpeg, tiff, pdf or xml files – this is subjected to segmentation and Extract, Transform and Load (ETL) operations using software developed in Java and XML to obtain article level data. This article level data with cleaned headlines are stored in a PostgreSQL database at the Center for Computational Learning Systems (CCLS). This serves as the backend for all our development. It took approximately 296 hours to upload all the data onto the servers – this included processing a total of 3361 files of which 780 were page level xml files and 60 were issue level xmls. The ER-diagram for the database is illustrated in Figure 5.1 in appendix. The Issue-Page-Article-Relator (IPAR id) acts as a unique key for each newspaper article.

In addition to the storage of OCR data from the newspaper articles, the database is also capable of storing version and change tracking since the patrons of the archive are expected to edit an article multiple times correcting garbled OCR and add tags and keywords. Thus, the data uploaded in the database will be subject to frequent updating, and all versions of the data are required to be stored by the database. Two approaches were explored to address the problem – create an

1. XML database and
2. Relational database with timestamps.

To test the memory requirements for the above approaches, three articles were chosen randomly and each line of these articles were copied and stored as a new version of the line, in both databases. In the XML based approach, an XML file was created for each page of the issue. Each node of the document had attributes which could uniquely identify the changes made to a text of line.

For the relational database a new table was created with columns to uniquely identify each text line, the user responsible for text changes and the timestamp when the change was made were stored. The memory requirements of both these approaches were analyzed. 212 lines of text were stored in each table. The XML file occupied 29.2 KB of memory, which indicates an average of 0.1377 kilobytes of memory per line of text. The relational version table occupied 24 KB, which indicates an average of 0.1132 kilobytes per line of text. Furthermore, loading a relational database is faster than an XML database. As the relational database requires roughly 18% less memory this was the preferred choice for the BODHI system.

Two XML database systems were also examined for feasibility study - MarkLogic server[2] and eXist[3]. However, the RDBMS approach was preferred due to the complexities and learning curve associated with establishing a full-fledged NativeXML database.

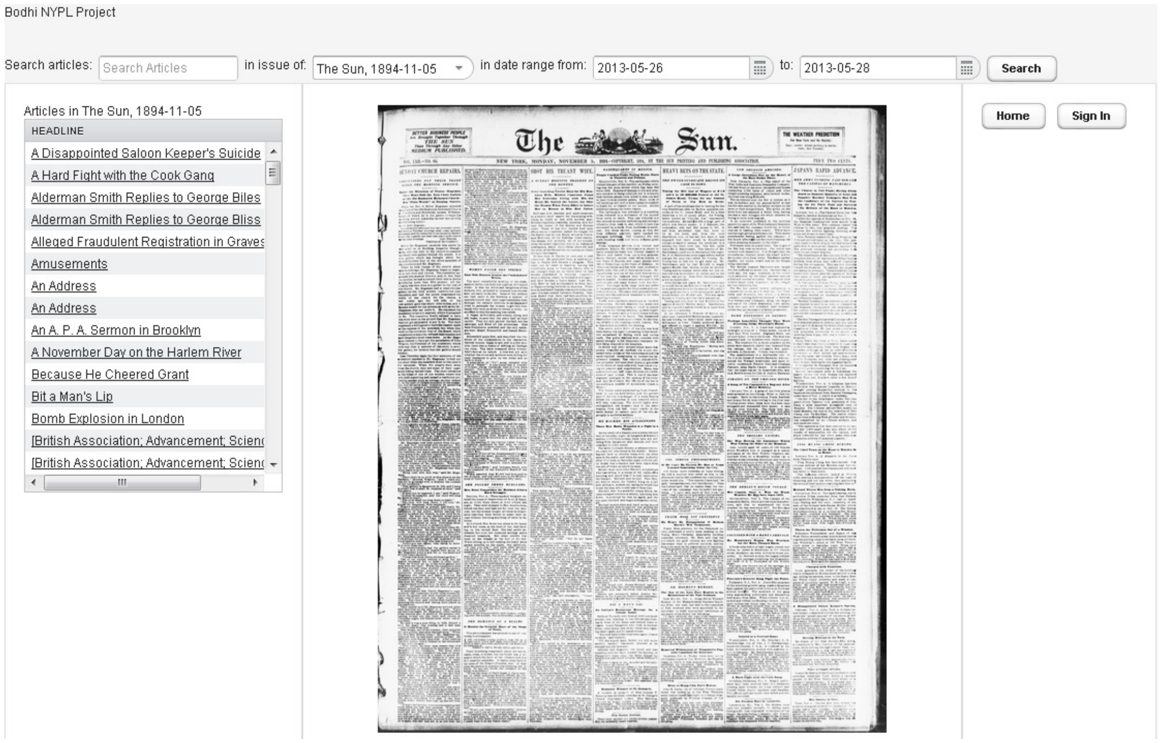
In addition to the storage of OCR data from the newspaper articles, the database is also capable of storing user registration information and is indexed using Apache Lucene (`http://lucene.apache.org/java/docs/index.html`), which has an open-source Java-based indexing and search implementation as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities.

3.0.3 The UI Design

The UI components of the BODHI application include a user manager, article display, OCR corrector, tag (or annotation) collector and search and retrieval manager. The prototype has been developed using Ruby-on-Rails and will be deployed using the Apache Phusion Passenger framework (`http://www.modrails.com/`). Figure 3.2 shows the interface developed. Once deployed on the library infrastructure, the web interface can be accessed by social media, mobile apps and the internet.

[2]http://www.marklogic.com/

[3]http://exist-db.org/exist/apps/homepage/index.html


Bodhi NYPL Project

Search articles: Search Articles in issue of: The Sun, 1894-11-05 in date range from: 2013-05-26 to: 2013-05-28 Search

Home Sign In

Articles in The Sun, 1894-11-05

HEADLINE
A Disappointed Saloon Keeper's Suicide
A Hard Fight with the Cook Gang
Alderman Smith Replies to George Biles
Alderman Smith Replies to George Bliss
Alleged Fraudulent Registration in Graves
Amusements
An Address
An Address
An A. P. A. Sermon in Brooklyn
A November Day on the Harlem River
Because He Cheered Grant
Bit a Man's Lip
Bomb Explosion in London
[British Association; Advancement; Scienc
[British Association; Advancement; Scienc

Figure 3.2: The User Interface for BODHI.

Chapter 4

Topic Modeling

The newspaper articles stored in the database need to be categorized to make search and retrieval efficient. Automatic categorization of articles from this archive is challenging – newspapers are scanned on a page-by-page basis and article level segmentation is poor or non-existent; the OCR scanning process is far from perfect and documents generated from it contain a large amount of garbled text. The OCR software provides a high level categorization of articles, but this is often not very helpful – for instance, an attempt to categorize articles in the edition of *The Sun* newspaper published on November 4th, 1894 resulted in 338 of the 413 articles of the newspaper (slightly more than 80%) classified as editorial.

Thus articles dealing with elections and governmental appointments, crime and public health are all assigned to the same category – there is no easy way to group these articles into meaningful sub-categories. Annotators were employed in our study (staff and students at CCLS and Amazon Mechanical Turkers) to understand whether a broad level categorization of the articles can be found. The sub-categories identified are highly subjective and so evaluation of annotator performance (and hence the categorization) is non-trivial.

4.1 Ground Truth

There is no oracle available to provide ground-truth regarding the newspaper article categorization. To establish ground truth, a similarity graph from the articles is built and

Community Structure Detection (CSD) algorithms [Newman, 2006] are run on it. This process partitions the graph into meaningful sub-categories. Each article forms a vertex of the graph and an edge exists between two vertices if the cosine similarity between the term frequency-inverse document frequency between articles exceeds a user-defined threshold. The choice of different thresholds naturally generate *multiple ground truths*. This method of generating ground truths takes into consideration the content of the articles, instead of the popular majority voting scheme where labels provided by annotators (and not the content) are considered for generating ground truth [Parent and Eskenazi., 2010].

In future work, we are also considering generative topic models [Blei, 2011; Steyvers and Griffiths, 2006] for determining topics of articles. However, depending on parameters of the distributions and choice of priors, multiple ground truths may exist in these formulations also.

4.2 Community Structure Detection

Community structure detection has its roots in network theory [Newman., 2004]. It uses properties of a graph to find communities. A community is defined to be a group of nodes that are densely connected to each other, while sparsely connected to other nodes outside of the group.

Some algorithms for community structure detection are briefly described here: The oldest method is the minimum-cut algorithm [Flake *et al.*, 2004], where a graph is partitioned into K sets, usually of the same size, and split in a way where the number of inter community edges is minimized. A disadvantage to using this algorithm is that K is an externally defined parameter and is unrelated with the properties of the graph. A popular method is the Girvan-Newman method [Newman and Girvan., 2004] that uses the *betweenness* measure to split a graph. Betweenness favors edges that lie between communities and disfavors those that lie inside communities. A key idea is that if two communities are connected by only a few inter-community edges, then all paths through the graph from vertices in one community to vertices in the other must pass along one of those few edges. Given a suitable set of paths, one can count how many go along each edge in the graph, and this number provides a measure of betweenness. The greater the value, the more likely the edge is an inter-community connection rather than an intra-community one. Another popular community structure detection method is modularity maximization, and this is used in our experiments.

4.2.1 Community Structure Detection Using Modularity Maximization:

A guiding principle for obtaining good division of a graph into communities is to find a partition in which there are few edges between communities; more importantly, the goal is to find different number of edges between communities compared to what is *expected* [Newman, 2006]. *Modularity* of a graph is defined to be the number of edges falling within communities minus the expected number in an equivalent graph with edges placed at random. A positive modularity value indicates intracommunity connections, while negative modularity indicates a lack of such connections.

Assume that a graph has n vertices; consider a partition of the graph into two groups – let $s_i = 1$ if vertex i belongs to group 1 and $s_i = -1$ if in group 2. The adjacency matrix of the graph can be represented by A_{ij} where $A_{ij} = 1$ if there is an edge between vertices i and j and 0 otherwise. The expected number of edges between vertices i and j if edges are placed at random is $\frac{k_ik_j}{2m}$ where k_p represents the degree of vertex p and $m = \frac{1}{2}\sum_p k_p$ is the total number of edges in the network. Thus the modularity can be written as $Q = {}_{1}4m \sum_{ij}(A_{ij} - \frac{k_ik_j}{2m})s_is_j = \frac{1}{4m}s^TBs$. The matrix $\mathbf{B}$ is called the *modularity matrix* and is similar in spirit to the *graph Laplacian* [Chung., 1997]. The s matrix can be written as a linear combination of normalized eigen vectors u_i of $\mathbf{B}$ so that $s = \sum_i a_iu_i$ with $a_i = u^Ts$. Then, $Q = \sum_i a_iu^TB\sum_j a_ju_j = \sum_i^n(u_i^T)^2\beta_i$ where β_i is the eigen value of B corresponding to the eigen vector u_i. If the eigen values are labeled in decreasing order $\beta_1, \beta_2, \cdots \beta_n$, one can maximize the modularity by choosing an appropriate division of the graph or the appropriate value of the vector s. The two constraints for maximizing the modularity are that s should be chosen such that as much weight as possible resides in the terms of the sum involving the largest (most positive) eigenvalues and the values of s should range in $s = \pm 1$. This makes it an NP-hard MAXCUT problem and approximate methods need to be designed to solve it. The algorithm used for approximating the modularity works by computing the leading eigen vectors of the modularity matrix and dividing the vertices into two groups based on the sign on the vector u_i. To extend the partitioning scheme to more than two groups, we use the above algorithm to first divide into two groups, then divide those two groups and so forth. More details about the algorithm is available from work done by Newman [Newman, 2006].

The choice of this algorithm was based on the fact that it compares the structure of a graph with that of a random graph. This important property can be used effectively to detect "bad" workers. The idea is that a graph generated from the work of a "bad" worker who randomly labels articles should be closer in likeness to a random graph than a legitimate annotator's work.

4.3 Second Opinion

Annotators provide noisy labels [Raykar and Yu., 2011]. Let y_i^j be the label provided by the j-th annotator to the i-th instance. The labels can be categorical and it is assumed that there are k such categories i.e. y_i^j can take values between $c_1, c_2 \cdots c_k$. The ground truth labels are given by $y_i^{g[p]}, 1 \leq p \leq m$, assuming m ground truths can be defined. The annotators help learn each ground-truth independently. If $y_i^j = c_k$ when $y_i^{g[p]} = c_k, p \in [1, m]$, then the annotators labels match with a realization of the ground truth.

4.3.1 Case Study 1: Unpaid Annotators

A pilot study is conducted to test whether the category labeled "editorial" by the OCR software could be further broken down to more meaningful sub-categories. Six annotators (primarily graduate students and researchers) are recruited to determine the number of natural categories found in a random sample of twenty-five articles. The articles (all labeled "article/editorial" by the OCR software) are selected from the November 2nd, 1894 issue of *The Sun* newspaper. The OCR software when scanning these articles created a lot of garbled text. Figure 4.1 shows the proportion of OCR tokens which were English words. All the annotators are given the same set of articles to work with. They are asked to skim the articles first and group them into obvious and intuitive categories, focusing on the

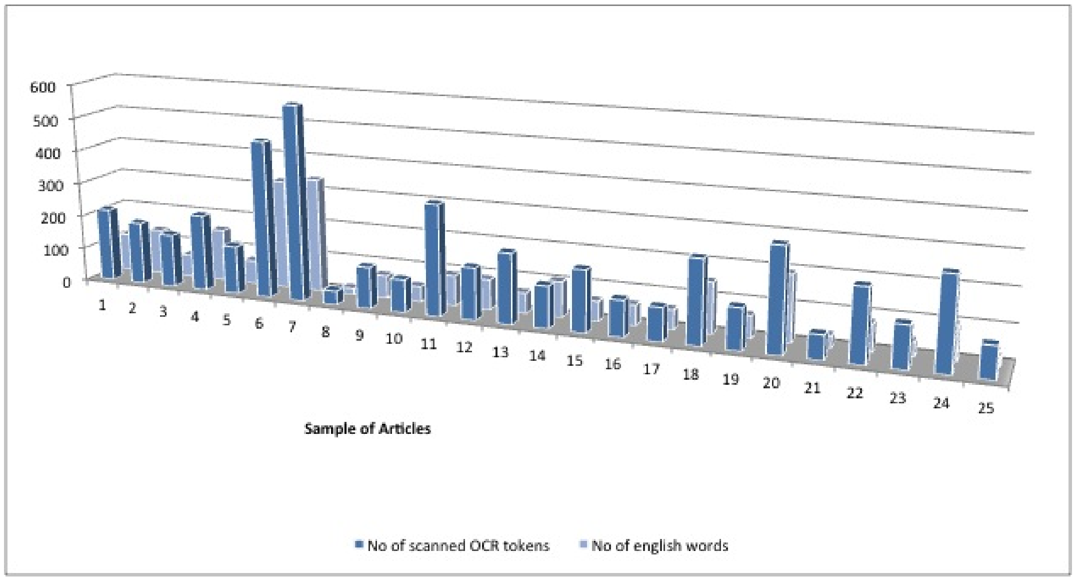


Figure 4.1: Number of english words and OCR tokens scanned for the 25 articles used in Case Study 1.

"bigger picture". The defined categories had be described in 5 - 10 words and preferably had to include words from the articles. They are also interviewed with the following set of questions: (1) What was the strategy you used for coming up with the categories? (2)Were there any documents that you found difficult to assign to categories? (3) Did you find any part of the study particularly difficult or ambiguous? If so, describe the problem you faced. (4) How long did it take you to complete the study? (5) If you had the opportunity to change anything with this study, what would it be? It is found that most annotators suggested a two step process – skimming through the articles once, focusing on the headlines and first paragraph and trying to summarize it or looking for criteria to associate pairs of articles; a second reading was necessary to pick topic words consistent with the high level category or regrouping articles as necessary.

Some sub-categories are very small or singletons while others contained close to 30% of the articles in the sample. Annotators suggested including a more diverse set of articles for the task which is being considered in a large scale study currently being designed. Some articles fit into more than one category, so there was some confusion on whether to allow

ID	Number of sub-categories found
Annotator 1	8
Annotator 2	13
Annotator 3	13
Annotator 4	9
Annotator 5	10
Annotator 6	13

Table 4.1: Sub-Categories found by humans in the random sample of 25 articles labeled "editorial" by the OCR software.

multiple categories per article. For this study, we emphasized the need to assign one article to one category.

Interpreting Results from the Pilot Study: Table 4.1 shows the suggested number of sub-categories found by the annotators. From this high level grouping, we further processed the labels to come up with six categories – article counts are illustrated in Figure 4.2. The November 2nd, 1894 newspaper was published immediately after general elections; thus a lot of articles in this issue had to do with politics and elections. Annotators unanimously agreed that seven of the twenty five articles used for the study belong to the category "politics/elections/governmental appointments". Three of the annotators found hierarchies among this category such as "politics/ballot, politics/election, politics/nomination, politics/war, politics/social, politics/ entertainment, politics/gossip". This accounted for the increased number of total categories they listed. Annotators merged together articles that contained arts, biographies, book reviews and the like into one category called "arts/human interest". Creating a homogeneous category for these articles was not easy due to the wide variety of articles. Articles pertaining to "death" and "marriage announcements" were binned into separate categories. There was no agreement among annotators on eleven articles – for example, an article with a headline "President Cleveland goes hunting for squirrels" was labeled as belonging to the following categories: human interest/politics/sports/ entertainment/social. All of these eleven articles had a much higher level of ambiguity and there was no agreement among annotators.

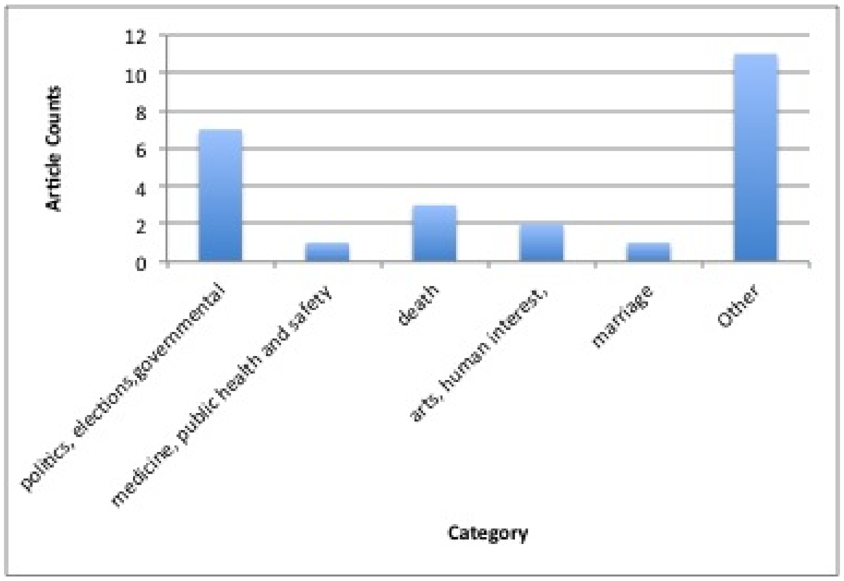


Figure 4.2: Counts of articles in each sub-category defined by annotators in the random sample of 25 articles.

The interview section of the annotation task indicated that small or singleton categories lead to less agreement among humans; these outliers do not fit into a larger category easily and this raised confusion and difficulty in categorization. Thus, learning from more examples of similar kind was the norm.

4.3.2 Case Study 2: Mechanical Turkers as Annotators

We designed an annotation experiment using Amazon's Mechanical Turk[1] (AMT). The twenty five articles sampled from the archive were broken down into groups of five and a Human Intelligence Task (HIT) presented to a Mechanical Turker comprised of labeling all five articles[2]. Annotators were explicitly asked to chose only one word representing the high level category for each article. 30 annotators were requested to provide their

[1]`https://www.mturk.com/mturk/welcome`

[2]This design of the HIT was chosen to ensure that each HIT had a low cognitive load for the workers. Cognitive load is a function of the number of articles in a HIT [Parent and Eskenazi., 2010].

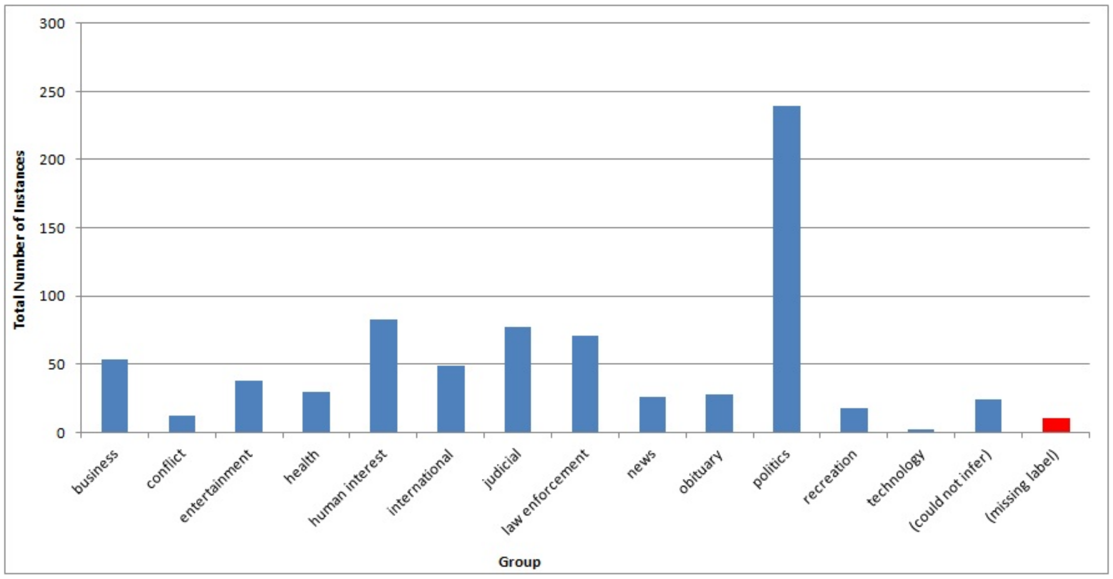
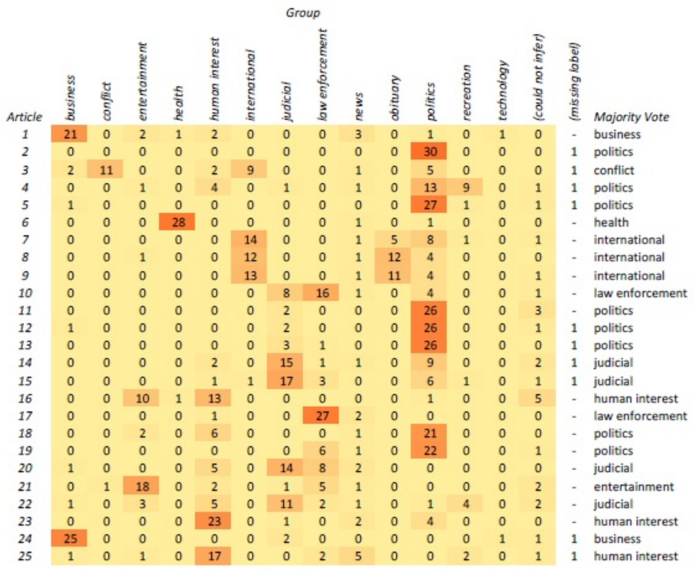


Figure 4.3: Counts of articles in sub-categories from the Amazon Mechanical Turk experiment. Some words (or group of words) could not be assigned a single category label and they have been grouped under "could not infer". A small percentage, shown in red, had missing labels.

feedback on each article. The results revealed that there were a total of 333 unique labels provided by annotators for the 25 samples of which 34 labels comprised of two words, 29 three words, 10 four words, 4 five words and 4 cases when the label was greater than seven words. This was in spite of the fact that explicit instructions were given to use a single word for labels. Thus raw labels provided by annotators needed to be pre-processed and grouped further. Misspellings (such as administation, bicycle herdles, condolance, suspention etc) were removed by manual examination. Table 1 in the Appendix presents examples of how words provided by annotators are mapped to high level categories. Fourteen categories emerged from words suggested by the Mechanical Turkers[3]. Figure 4.5 shows the distribution of the words suggested as labels by annotators. Interestingly, the most popular word was "politics" which was suggested as a label 140 times.

[3]It must be noted that there is no unique way of assigning words suggested by annotators to categories. The empirical results presented here are simply one possible way of aggregating suggested words.

Because of the way we set up the HITs, we could not directly compare the AMT annotators with the six original annotators. Instead, we created a *pseudo-annotator*(denoted by P in subsequent discussions) based on the results of the Turkers. The pseudo-annotator labeled an article based on the majority vote of 30 annotators who tagged each article. For nine of the articles, there was a strong agreement regarding the category. In each, at least 25 annotators voted the same way. Seven other articles had fairly strong agreement, which we defined as having at least 15 majority votes and at least twice as many majority votes as the secondary vote. The remaining nine articles showed disagreement among the annotators. But even here, they were only split between two categories. In these instances, the pseudo-annotator selected the category with the most votes. Overall, the pseudo-annotator only used nine categories instead of all fourteen. The AMT annotators took anywhere from 21 seconds to 17 hours to complete the tasks.



	Group															
Article	business	conflict	entertainment	health	human interest	international	judicial	law enforcement	news	obituary	politics	recreation	technology	(could not infer)	(missing label)	Majority Vote
1	21	0	2	1	2	0	0	0	3	0	1	0	1	0	-	business
2	0	0	0	0	0	0	0	0	0	0	30	0	0	0	1	politics
3	2	11	0	0	2	9	0	0	1	0	5	0	0	0	1	conflict
4	0	0	1	0	4	0	1	0	1	0	13	9	0	1	1	politics
5	1	0	0	0	0	0	0	0	0	0	27	1	0	1	1	politics
6	0	0	0	28	0	0	0	0	1	0	1	0	0	0	-	health
7	0	0	0	0	0	14	0	0	1	5	8	1	0	1	-	international
8	0	0	1	0	0	12	0	0	1	12	4	0	0	0	-	international
9	0	0	0	0	0	13	0	0	1	11	4	0	0	1	-	international
10	0	0	0	0	0	0	8	16	1	0	4	0	0	1	-	law enforcement
11	0	0	0	0	0	0	2	0	0	0	26	0	0	3	-	politics
12	1	0	0	0	0	0	2	0	0	0	26	0	0	1	1	politics
13	0	0	0	0	0	0	3	1	0	0	26	0	0	0	1	politics
14	0	0	0	0	2	0	15	1	1	0	9	0	0	2	1	judicial
15	0	0	0	0	1	1	17	3	0	0	6	1	0	1	1	judicial
16	0	0	10	1	13	0	0	0	0	0	1	0	0	5	-	human interest
17	0	0	0	0	1	0	0	27	2	0	0	0	0	0	-	law enforcement
18	0	0	2	0	6	0	0	0	1	0	21	0	0	0	-	politics
19	0	0	0	0	0	0	0	6	1	0	22	0	0	1	-	politics
20	1	0	0	0	5	0	14	8	2	0	0	0	0	0	-	judicial
21	0	1	18	0	2	0	1	5	1	0	0	0	0	2	-	entertainment
22	1	0	3	0	5	0	11	2	1	0	1	4	0	2	-	judicial
23	0	0	0	0	23	0	1	0	2	0	4	0	0	0	-	human interest
24	25	0	0	0	0	0	2	0	0	0	0	0	1	1	1	business
25	1	0	1	0	17	0	0	2	5	0	0	2	0	1	1	human interest

Figure 4.4: Mapping the categories provided by Mechanical Turkers to fourteen high-level categories.

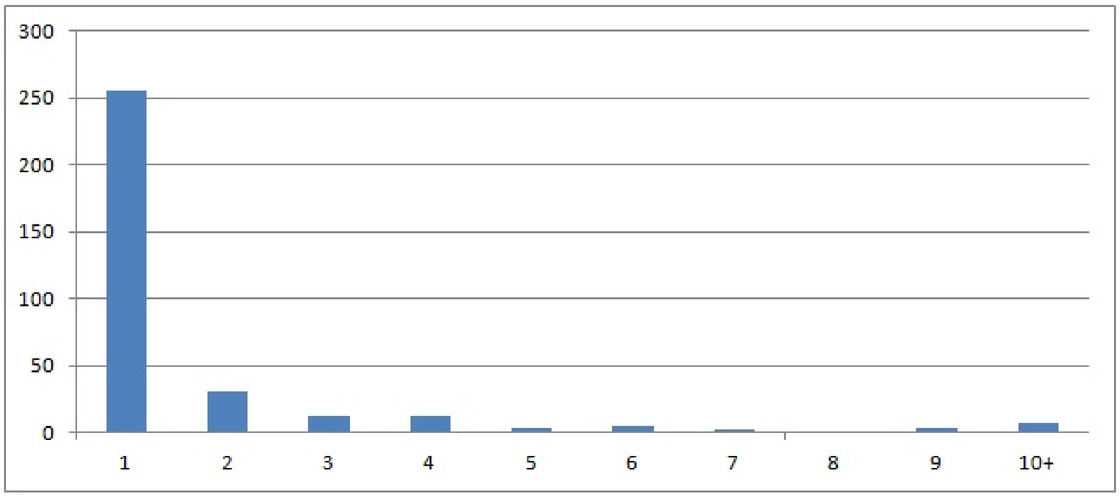
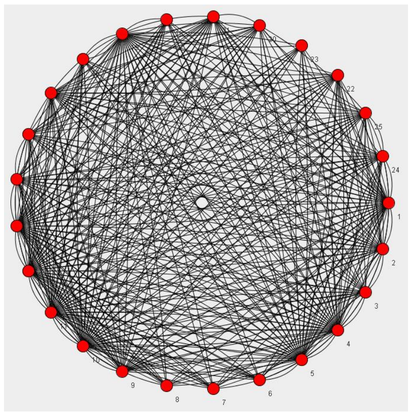


Figure 4.5: Distribution of words used as labels in the amazon mechanical turk experiment

4.4 Information Retrieval Metrics

Several information retrieval metrics can be used to compare annotators to the established gold standard. Let each data set D have n instances $O_1, O_2, \cdots, O_n$ and we want to partition it into K communities. Let $K = \{1, 2, \cdots \eta\}$ be the set of labels assigned by the community detection algorithm and $C = \{1, 2, \cdots, \eta\}$ be the expert annotated class labels assigned to the objects in D. (1) **Correlation Coefficient:** Correlation coefficient measures how closely the labels obtained from the community detection algorithm relates to the annotator provided labels. It is defined as: $\frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{(\sum(x_i-\bar{x})^2)(\sum(y_i-\bar{y})^2)}, x_i \in K, y_i \in C$ (2) **Mutual Information:** Consider a two-dimensional contingency table, $\mathcal{H} = h(c,k)$ where $h(c,k)$ represents the number of objects labeled class c are assigned to community k by the algorithm. Then, if there is a perfect matching $\mathcal{H}$ is a square matrix with only one non-zero element per row / column. The marginals are defined as $h(c) = \sum_k h(c,k)$ and $h(k) = \sum_c h(c,k)$. The empirical mutual information is measured by $\hat{I}(C,K) = \hat{H}(C) - \hat{H}(C|K)$, where $\hat{H}(C) = -\sum_{c=1}^{|C|} \frac{h(c)}{n} log \frac{h(c)}{n}$ and
$\hat{H}(C|K) = -\sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{h(c,k)}{n} log \frac{h(c,k)}{h(k)}$. (3) **Normalized Mutual Information:** The normalized mutual information is measured by: $\frac{I(C;K)}{\sqrt{H(K)H(C)}}$, where H is the entropy. (4)**Rand Index:** Let, TP = the number of pairs of elements in D that are in the same


(a) Threshold=0.10

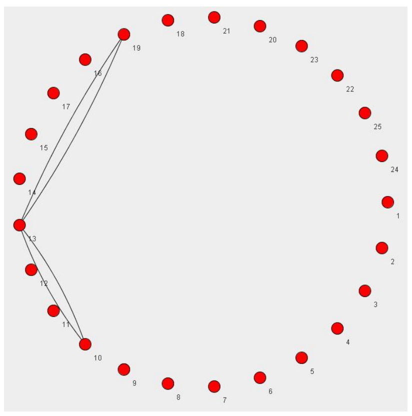

(b) Threshold=0.5

Figure 4.6: Annotations obtained from the community structure detection algorithm for two different thresholds of cosine similarity between articles.

set in C and in K. TN = the number of pairs of elements in D that are in different sets in C and in K. FP = the number of pairs of elements in D that are in the same set in C but in different sets in K. FN = the number of pairs of elements in D that are in different sets in C but in the same set in K. Then, the Rand index, $R = \frac{TP+TN}{TP+TN+FP+FN}$ (5) **Adjusted Rand Index:** This metric [Hubert and Arabie., 1985] factors out the portions of the two partitions that match purely by chance. One can build a contingency table of the elements in common between both partitions, $[n_{ij}]$, where n_{ij} is the number of elements that match in both clusterings, and a_i is the sum of all n_{ij} and b_j is the sum of all n_{ij}. Then, the Adjusted Rand Index is the following: ARI =

$$\frac{\sum_{ij}({}_{n_{ij}}\mathrm{C}_2) - [\sum_{ij}({}_{a_i}\mathrm{C}_2)\sum_{ij}({}_{b_j}\mathrm{C}_2)]/{}_n\mathrm{C}_2}{[\sum_{ij}({}_{a_i}\mathrm{C}_2) + \sum_{ij}({}_{b_j}\mathrm{C}_2)]/2 - [\sum_{ij}({}_{a_i}\mathrm{C}_2)\sum_{ij}({}_{b_j}\mathrm{C}_2)]/{}_n\mathrm{C}_2} \quad (4.1)$$

The index ranges between -1 and 1. (6)**Crammers V:** is the intercorrelation of two discrete variables with categorical values. It varies from 0 (corresponding to no association between the variables) to 1 (complete association) and can reach 1 only when the two variables are equal to each other. It is given by: $\sqrt{\frac{\chi^2}{N(k-1)}}$ where χ^2 is obtained from Pearson's chi-squared test, N is the total number of observations and k is the number of optional outcomes usually chosen as the smaller number of rows or columns.

	Annotator						
Th(#Cms)	1	2	3	4	5	6	P
0.10(9)	0.34	0.28	0.07	0.26	0.31	0.05	0.08
0.12(23)	-0.26	0.53	0.78	0.58	0.57	0.65	-0.17
0.13(10)	0.06	0.23	0.36	0.18	0.23	0.17	-0.09
0.14(17)	0.02	0.67	0.60	0.56	0.55	0.39	-0.16
0.15(5)	-0.36	0.11	0.14	0.09	0.22	0.24	-0.09
Ann Clus	8	14	12	9	10	9	9

Table 4.2: Correlation Coefficients Comparing Annotators to Community Structure Detection Results

4.5 Experimental Results

25 articles are sampled randomly from the archive and formed vertices of a similarity graph; thresholded cosine similarity between the term frequency-inverse document frequency is used to determine whether an edge should exist between two vertices.

Automatic Community Structure Detection: By adjusting the threshold[4] of cosine similarity, it is possible to create a large set of graphs with differing number of edges. The cosine similarity value in our application ranged from 0.03 to 0.5. Some graphs are ignored because they are duplicates of others – in all, there are twenty-three graphs we work with. Newman's algorithm (as described in Section 4.2) is run on each graph. Figure 4.6 shows two examples of the graphs obtained using different values of thresholds for the cosine similarity between articles. Many graphs resulted in only one or two communities. Although the articles can be grouped into just two categories, this is not a particularly interesting result. Most users are likely to prefer a higher number of communities, so the analysis is restricted to those that produced more than four communities. An upper bound on the cosine similarity is also imposed to remove any thresholds that produced graphs with isolated nodes. We study five graphs closely, produced by setting thresholds at 0.10, 0.12, 0.13, 0.14, and 0.15. Each of these graphs can be potentially considered as "ground

[4]If the cosine similarity is above the threshold, an edge exists between vertices, otherwise not.

truth" and the question of which one to use is left to the application designer and is not the focus of the discussion here; instead, we study the performance of the annotators using any "ground truth" as a baseline.

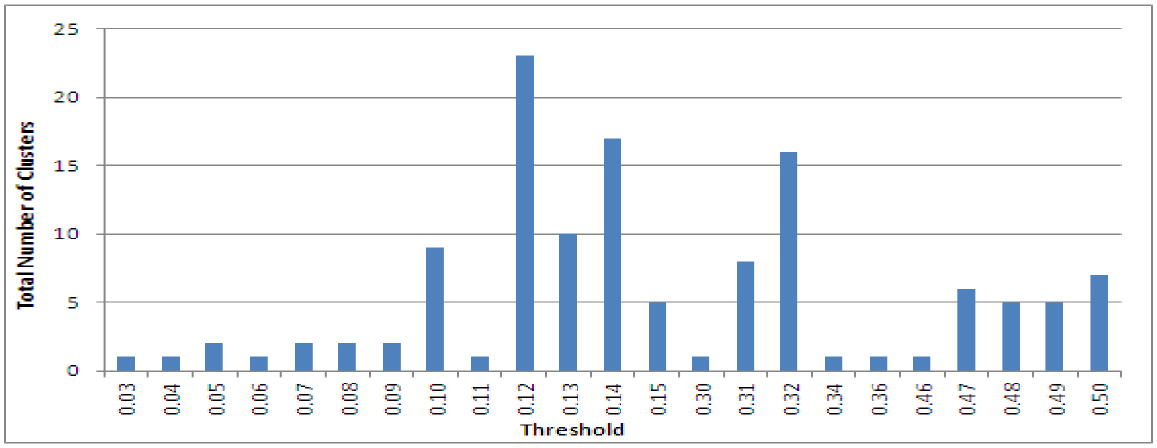


Figure 4.7: Plot of the Number of Clusters for different thresholds of the cosine similarity.

Annotations from Humans: Figure 1 (in Appendix) shows the graphs produced by the six annotators and the pseudo-annotator from the AMT experiment[5].

Which metric to use? The information retrieval metrics described in section 4.4 are applied on the labels obtained by running the CSD algorithm (for all the above mentioned thresholds) versus those suggested by annotators. To help decide which metric is most useful in ranking annotators, we calculate *range* of a metric as the difference between the maximum and minimum scores obtained across all annotators at a given threshold. For example, in Table 4.2, for correlation coefficient at threshold 0.13, the spread of the correlation coefficient is $0.36 - (-0.09) = 0.45$. The sum of range over all thresholds is the total range. The metric with highest total range is deemed most suitable. Intuitively, this metric allows large separation of scores assigned to good and bad annotators. The correlation coefficient had the largest range (1.3114) followed by Crammer's V (1.000). Figure 4.9 shows a comparison of metrics obtained by fixing the threshold of cosine similarity to 0.14 in the CSD algorithm.

Ranking annotators: Sorting by raw scores from correlation coefficient, the follow-

[5]In this work, the Java Universal Network/Graph Framework[6] has been used for graph visualization.

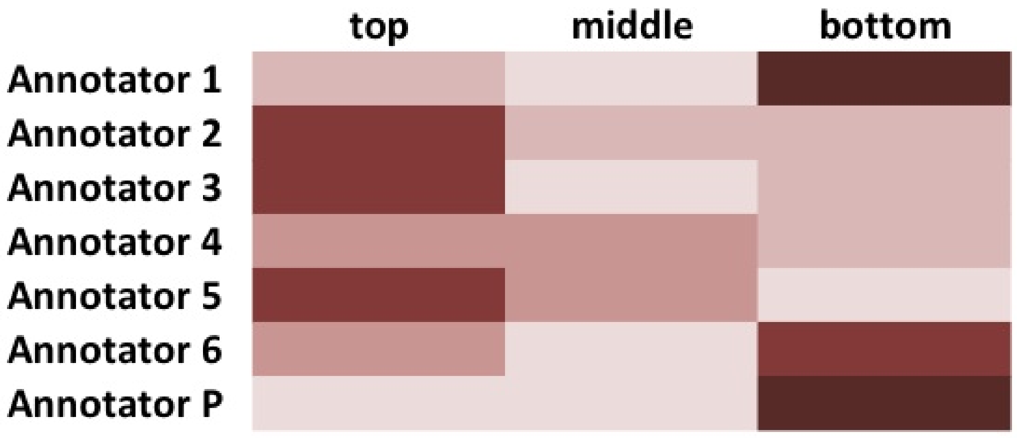
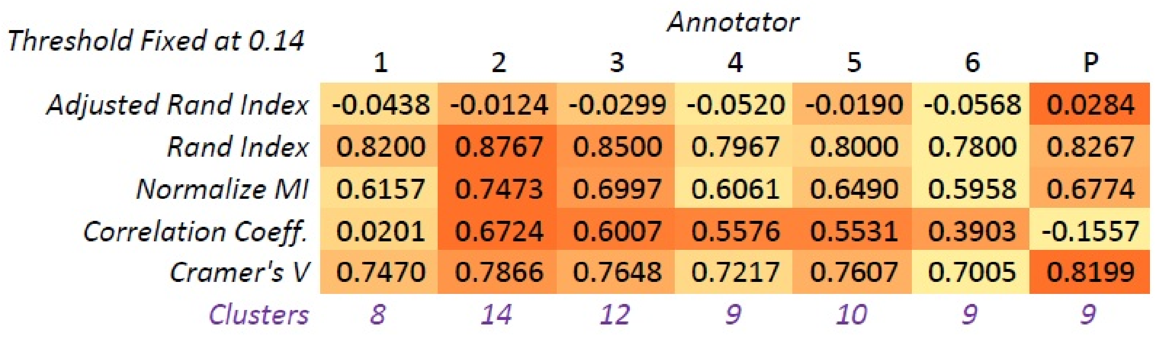


Figure 4.8: Ranking of annotators using the correlation coefficient metric. The color map is designed such that 0 has the lightest shade and 4 the darkest. Top, middle and bottom refer to the position in the ranked list. A dark color corresponds to many occurrences in the corresponding part of the list.

ing rankings are observed: at threshold 0.10, the rank is $1, 5, 2, 4, P, 3, 6$; at 0.12 it is $3, 6, 4, 5, 2, 1, P$; at 0.13 it is $3, 5, 2, 4, 6, 1, P$; at 0.14 it is $2, 3, 4, 5, 6, 1, P$ and at 0.15 it is $6, 5, 3, 2, 4, P, 1$. If an annotator has any one of the top three ranks, we assume s/he is on the top of the list, a rank of four corresponds to the middle and the remaining three the bottom of the list. Based on this categorization, the heat map shown in Figure 4.8 is obtained. Overall, annotators $2, 3, 5$ appear to be on top of the list while $1, P$ are consistently bad workers. This implies that majority voting may not be a reliable metric for choosing annotators for find sub-categories. It is important to note though, the ranking is subjective and depends heavily on the choice of the "ground truth". As such, some workers are better than others at certain tasks, but it is not straight forward to generalize this observation.

A high correlation coefficient value occurred with the graph of threshold 0.14 (Figure 4.9). If the ground truth is fixed at this threshold, annotators 2 and 3 produced graphs that were most similar to the CSD graphs (14 and 12 clusters compared to 17 in CSD at threshold 0.14) and have the highest correlation coefficient values. The other annotators only created 8 to 10 clusters and had lower values.



Threshold Fixed at 0.14	Annotator 1	2	3	4	5	6	P
Adjusted Rand Index	-0.0438	-0.0124	-0.0299	-0.0520	-0.0190	-0.0568	0.0284
Rand Index	0.8200	0.8767	0.8500	0.7967	0.8000	0.7800	0.8267
Normalize MI	0.6157	0.7473	0.6997	0.6061	0.6490	0.5958	0.6774
Correlation Coeff.	0.0201	0.6724	0.6007	0.5576	0.5531	0.3903	-0.1557
Cramer's V	0.7470	0.7866	0.7648	0.7217	0.7607	0.7005	0.8199
Clusters	8	14	12	9	10	9	9

Figure 4.9: Comparison of metrics using a fixed threshold for the CSD algorithm.

4.6 Graphical Models for Community Structure Discovery in Document Networks

4.6.1 Terminology

We are in the process of studying generative models for community structure discovery in document networks. The problem formulation is similar to the LDA-based mixture model for social networks described in [Zhang *et al.*, 2007]. Figure 4.10 illustrates the graphical model that can be used to represent topic communities from document networks, assuming that the number of communities is known before hand.

Let G represent a document network comprising of a set of documents $V = \{v_1, v_2 \cdots v_M\}$ linked to each other by a set of edges $E = \{e_1, e_2, \cdots, e_N\}$. Each document is represented by a term frequency - inverse document frequency (tf-idf) vector and the cosine similarity between them is evaluated. An edge exists between the two documents if the cosine similarity is greater than an user-defined threshold. The cosine similarity is also used as the interaction weight function $IW : (V \times V) \rightarrow I$.

Assume that the document v_i has a set of neighbors $\vec{\omega_i}$ such that ω_{ij} represents the j^{th} neighbor of the document i. Each document is characterized by an Interaction Profile (IP) which is defined as follows: $IP(v_i) = \{(\omega_{i1}, IW(v_i, \omega_{i1})), \cdots, (\omega_{im_i}, IW(v_i, \omega_{im_i}))\}$ where m_i represents the set of neighbors corresponding to the document v_i in the network. The interaction between the documents is considered exchangeable and this assumption permits application of an LDA-like model.

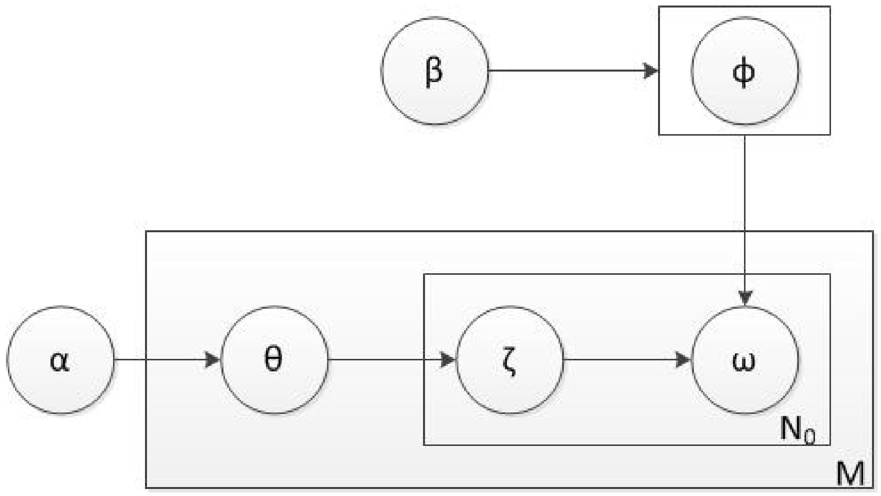


Figure 4.10: Graphical Model representing the topic communities. This formulation assumes that the number of communities is known before hand. Since each annotator provides labels for the documents in the archive, a graphical model similar to the one shown here can be designed representing probabilistic communities from document networks.

The document network G contains a set of communities $\zeta = (\zeta_1, \zeta_2, \cdots, \zeta_K)$ and each community in ζ is a distribution on the document set. The communities are modeled as latent variables in the graphical model and the cosine similarity between the documents are the visible variables. The community proportion variable θ is regulated by a Dirichlet distribution with a known parameter α. Each document belongs to every community with different probabilities and its interaction profile can be represented as a random mixture over latent community variables.
The distribution of topics in documents and terms in documents are two multinomial distributions with two Dirichlet priors, whose hyperparameters are $\vec{\alpha}$ and $\vec{\beta}$ respectively. The generative process for a node i's interaction profile in the document network is described

M	Number of documents in the network
K	Number of communities
N_i	Neighbor set for document i
$\vec{\alpha}$	Dirichlet prior hyperparameter on the mixing proportion
$\vec{\beta}$	Dirichlet prior hyperparameter on mixture component distributions
ζ	set of hidden community variables
$\vec{\theta}$	$p(\zeta\|IP(v_i))$ the community mixture proportion for $IP(v_i)$
Θ	$\{\vec{\theta_m}\}_{m=1}^{M}$
$\vec{\phi_k}$	$p(\omega\|zeta_K)$ the mixture component of community K
Φ	$\{\vec{\phi_k}\}_{k=1}^{K}$ estimated parameter set for community mixture
$\vec{\omega_i}$	the set of neighbors of document i

Table 4.3: Notation used in the graphical model.

as follows:

1. Sample mixture components $\vec{\phi_k} \sim Dir(\vec{\beta})$ for $k \in [1, K]$

2. Choose $\vec{\Theta_i} \sim Dir(\vec{\alpha})$ 3. Choose $N_i \sim Poisson(\epsilon)$ 4. For each of the N_i social interactions w_{ij}

(a) Choose a community $\zeta_{ij} \sim Multinomial(\vec{\theta_i})$

(b) Choose a social interaction $w_{ij} \sim Multinomial(\vec{\phi_{\zeta_{ij}}})$

According to the model, the probability that the j_{th} interaction element ω_j biases a neighbor ω_m is given by $p(\vec{\omega_i}, \vec{\zeta_i}, \vec{\theta_i}, \Phi|\vec{\alpha}, \vec{\beta}) = \prod_{j=1}^{N_i} p(\omega_{ij}|\vec{\phi_{\zeta_{ij}}})p(\zeta_{ij}|\vec{\theta_i})p(\theta_i|\vec{\alpha})p(\Phi|\vec{\beta})$

Exact inference is generally intractable for the LDA model. There have been primarily three approaches to solving this problem – expectation propagation, Gibbs Sampling and variational expectation maximization. The project participants are exploring large scale expectation propagation algorithms for inference from this graphical model.

Chapter 5

Grant Products

The research led to a number of publications including:

- Austin Lee, Haimonti Dutta, Rebecca Passonneau, David Waltz and Barbara Taranto, "Topic Identification from Historic Newspaper Articles of the New York Public Library: A Case Study", 5th Annual Machine Learning Symposium, New York Academy of Sciences, 2010.
- Haimonti Dutta, Rebecca J. Passonneau, Austin Lee, Axinia Radeva, Boyi Xie, David Waltz and Barbara Taranto, "Learning Parameters of the K-Means Algorithm from Subjective Human Annotation.", The 24th International FLAIRS Conference, Special Track on Data Mining, Palm Beach, FL. May 18-20, 2011.
- Haimonti Dutta and William Chan. "Using community structure detection to rank annotators when ground truth is subjective", NIPS Workshop on Human Computation for Science and Computational Sustainability, Lake Tahoe, December 7th, 2012.
- Haimonti Dutta, William Chan, Deepak Shankargouda, Manoj Pooleery, Axinia Radeva, Kyle Rego, Boyi Xie, Rebecca Passonneau, Austin Lee, Barbara Taranto. "Leveraging Subjective Human Annotation for Clustering Historic Newspaper Articles", Technical Report, arXiv:1208.3530.

Prototype software for OCR correction and tagging has also been developed which will be deployed on the CCLS servers and discussed with the New York Public Library for testing and use by patrons.

Part I

Appendices

5.1 Appendix

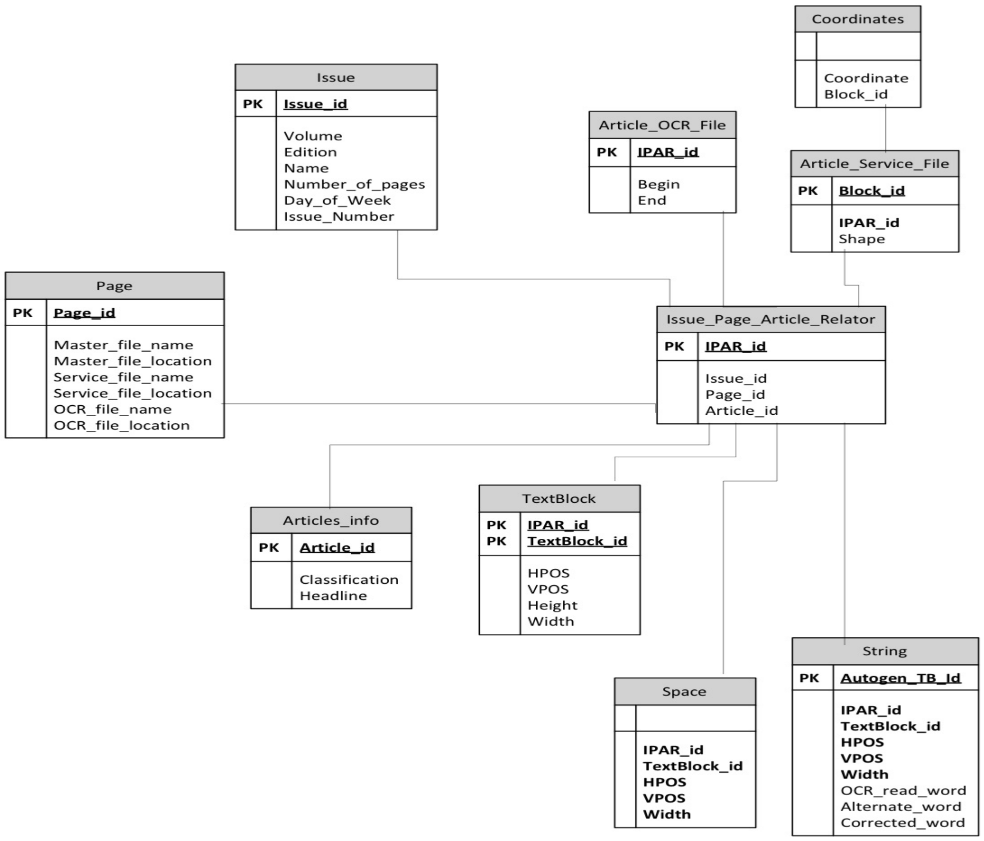


Figure 5.1: The ER Diagram of the database used to store the newspaper articles with cleaned headlines at CCLS.

Part II

Bibliography

Bibliography

[Blei, 2011] David M. Blei. Introduction to probabilistic topic models. *Communications of the ACM*, 2011.

[Chung., 1997] F. R. K. Chung. *Spectral Graph Theory*, volume 92. 1997.

[Flake *et al.*, 2004] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1:385–408, 2004.

[Hubert and Arabie., 1985] L. Hubert and P. Arabie. Comparing partitions. 2:193–218, 1985.

[Newman and Girvan., 2004] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, Feb 2004.

[Newman., 2004] M. Newman. Detecting community structure in networks. The European Physical Journal B - Condensed Matter and Complex Systems, 38:321–330, 2004.

[Newman, 2006] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[Parent and Eskenazi., 2010] Gabriel Parent and Maxine Eskenazi. Clustering dictionary definitions using amazon mechanical turk. *In Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, Association for Computational Linguistics*, 21-29, 2010.

[Raykar and Yu., 2011] V. C. Raykar and S. Yu. Ranking annotators for crowdsourced labeling tasks. pages 1809–1817, 2011.

[Steyvers and Griffiths, 2006] Mark Steyvers and Tom Griffiths. Probabilistic topic models. In T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning.* Laurence Erlbaum, 2006.

[Zhang *et al.*, 2007] H. Zhang, Baojun Q., C.L. Giles, H. C. Foley, and J. Yen. An lda-based community structure discovery approach for large-scale social networks. In *Intelligence and Security Informatics, 2007 IEEE*, pages 200–207, 2007.